Synergy-based Policy Improvement with Path Integrals for Anthropomorphic Hands

Fanny Ficuciello, Damiano Zaccara, Bruno Siciliano

Abstract— **In this work, a synergy-based reinforcement learning algorithm has been developed to confer autonomous grasping capabilities to anthropomorphic hands. In the presence of high degrees of freedom, classical machine learning techniques require a number of iterations that increases with the size of the problem, thus convergence of the solution is not ensured. The use of postural synergies determines dimensionality reduction of the search space and allows recent learning techniques, such as Policy Improvement with Path Integrals, to become easily applicable. A key point is the adoption of a suitable reward function representing the goal of the task and ensuring one-step performance evaluation. Force-closure quality of the grasp in the synergies subspace has been chosen as a cost function for performance evaluation. The experiments conducted on the SCHUNK 5-Finger Hand demonstrate the effectiveness of the algorithm showing skills comparable to human capabilities in learning new grasps and in performing a wide variety from power to high precision grasps of very small objects.**

I. Introduction

New generation of robots, to serve humans by substituting them in any kind of application or also by replacing parts of the body, should have comparable abilities to deftly move in different environments, autonomously learn and make decisions. To learn new tasks just as humans do, i.e. through trial-and-error policy, physical interaction is crucial. Therefore, advanced mechatronic structure and high number of degrees freedom (DoFs) for a robot are essential to change different configurations and adapt to the environment. At the same time, design and control complication due to high DoFs can be somewhat offset by means of coordinated motion patterns and sensory-motor synergies that help to simplify robot hardware and software [1]. This can be summarized by saying that the robot must be equipped with embodied intelligence.

This work focuses on one of the most fascinating and complex part of human and robot body in terms of mechanical design, sensors and control, namely the hand. To play the same role and functions of the human hand, artificial hands require anthropomorphic design, human-inspired control strategies and autonomous learning from interaction and exploration of the environment. Grasp synthesis based on analytic approaches suffers from computational complexity and modelling difficulties. First of all a precise model of the object should be available and, even more complex, a task description is needed to model object affordance. For this reason, new research trends in robotics go toward learning strategies that can integrate model-based pre-programmed actions with real-time learning from actions [2]. In the literature different classifications of learning strategies for robotics exist [3], [4]. In [5] two broad categories are distinguished for learning grasping mainly on the basis of the focus of observation, i.e. techniques centered on the observation of humans performing the grasp and techniques centered on the observation of the grasped object. To the first category belongs learning-by-demonstration strategies where sensors and signal processing are the key points for categorization [6], [7]; to the second category belong strategies that learn to associate grasp parameters to object geometric features or learn to identify grasping regions in an object image [8], [9]. The difference between imitation learning and Reinforcement Learning (RL) are highlighted in [10] and in particular a survey of RL in the context of robotics is provided. RL represents the future trends of learning strategies in robotics since it provides a robot with autonomous capabilities of learning new tasks on the basis of exploration and trial-and-error policy. Several RL approaches are found in robotics literature and are mainly based on policy search methods, which are preferable with respect to value function approaches as the latter are not suitable for high dimensional state and action space [11]. Different approaches for implementing policy-search are available with pros and cons, e.g. policy-gradient algorithms [12], Expectation-Maximization (EM) [13], yet more interesting results come from search algorithms from the field of stochastic optimization such as Policy Improvement with Path Integrals (PI^2) [14], [15]. This method belongs to the framework of stochastic optimal control and overcomes gradient computation of a cost function for the parameters update, thus avoiding problems related with discontinuities and noise in the cost functions. In this direction promising methods are obtained combining Cross-Entropy Method (CEM) optimization algorithm [16] to overcome a limit of PI^2 method that has a constant degree of exploration during the learning process. Hence, to obtain an exploration-exploitation trade-off, adaptive exploration is conceived by integrating in PI^2 the Covariance Matrix Adaptation (CAM) rule from the CEM [17], [18], [19]. In the field of RL some examples of application to robotics can be found in [20], [21], [22], [23]. The application on an anthropomorphic hand is realized in [24]. PI^2 is used to learn particular tasks involving two fingers such as slide a switch and turn a knob. The trajectories are represented via Dynamic Movement Primitives (DMPs), and are learned in the tendon-space of the index finger and of the thumb. The authors demonstrate

Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione, Università degli Studi di Napoli Federico II, via Claudio 21, 80125 Napoli, Italy, email: {fanny.ficuciello, bruno.siciliano}@unina.it, damianozaccara@libero.it. This research has been partially funded by the EC Seventh Framework Programme (FP7) within RoDyMan project 320992.

that optimizing a small number of trajectories in the synergy space can produce comparable performance to optimizing the trajectories of the tendons individually. In this work, the goal is quite different as well as the method used to define a policy, its initial parameters, and the reward function. The design of appropriate policy representations is essential for RL methods to be successfully applied to real-world robots. The idea is to demonstrate that a synergy-based approach is powerful for learning grasping with anthropomorphic hands due to configuration space dimesionality reduction. Motor synergies are a paradigm, inherited form neuroscience studies on the human hand [25], to represent joint couplings and inter-finger coordination as a powerful tool to plan grasps and control artificial hands using few parameters compared to the degrees of freedom (DOF) of the hand itself. Principal component analysis (PCA) and human grasps data set serve as data structures to define a policy and its initial parameters for a reinforcement learning algorithm. Indeed, starting from a "good enough" demonstration, the algorithm can optimize the policy parameters to gradually refine a stable grasp. When a clear measure about the success of the task is available, RL adaptability to new objects is ensured. The paper is organized as follows. In Sect. II the main characteristic of the Policy Improvement with Path Integrals algorithm adopted in this work is described. Section III provides the description of the hardware available for the experiments, i.e. the SCHUNK 5-Finger Hand; moreover, the description of the motor synergies subspace underlying the PI^2 policy is also provided; Section IV describes the reward function and the learning policy. In Sect. V experimental results are reported to validate the efficiency of the method in realizing grasps of objects with different shape and size. Finally, Section VI and VII respectively provide a discussion of the results and the conclusions.

II. Policy Improvement with Path Integrals (PI^2)

In reinforcement learning, the agent and its environment, such as a robot arm that inserts a peg in a hole or a mobile robot moving in a room or a hand that grasps objects, are modeled with a *state* $s \in S$ and can perform *action* $a \in A$. The action a changes the state of the system and the agent receives a feedback in terms of a scalar function named *reward* that measures the one-step performance of the robot with respect to the desired goal. The function π that maps states to actions is called *policy*. The goal of the RL is to discover the policy that maximizes the cumulative expected reward. Due to robotic systems high-dimensionality, Policy Search (PS) methods represent the optimal choice in robotics with respect to the classical RL techniques since the search space dimension is reduced by operating directly in the parameter space of the policy. Several policy search methods have been developed over the last two decades [10], [18], [19]. The Policy Improvement with Path Integrals algorithm is one of the most efficient and numerically robust examples of this approach and comes from the field of stochastic optimization. Unlike other policy search algorithm, PI^2 does not require a gradient estimate for the parameters update since it uses the principle of probability-weighted averaging to compute changes of the policy parameters, avoiding numerical instabilities due to matrix inversions. Minimizing a cost function through an iterative process of exploration and parameter updating is the goal of this method. The exploration is done by taking K samples $\boldsymbol{\theta}_{k=1\cdots K}$ from a Multivariate Gaussian distribution, with mean $\boldsymbol{\theta}$ and covariance matrix $\boldsymbol{\Sigma}$. The vector $\boldsymbol{\theta}$ represents the parameters of a policy $\pi\left(\boldsymbol{\theta}\right)$, which yields a specific trajectory $\boldsymbol{\tau}_k$. Each of these samples leads to different cost. The cost is determined by evaluating a task specific scalar function $S\left(\boldsymbol{\tau}_k\right)$, which is defined in terms of the trajectory since the reward depends on the robot performance. The trajectory assumes a different interpretation depending on the particular application. The expression of the adopted algorithm is provided in Algorithm 1 (see below) [14]. As previously discussed, the main difference between PI^2 and other policy search algorithms is the parameter update rule. In order to update the policy parameters, the PI^2 assigns to each trajectories a probability in inverse proportion to their rewards, as in line 8 of Algorithm 1.

This is the key point of the entire algorithm, since the new policy parameters are evaluated by performing probability-weighted averaging on the samples, as in line 10 of Algorithm 1. Therefore, the PI^2 method updates the parameter vector $\boldsymbol{\theta}$ such that it is expected to generate trajectories that lead to lower costs. The process continues with the new $\boldsymbol{\theta}$ as the basis for the new exploration. The classical PI^2 implementation provides only the distribution mean updating. Therefore, the exploration degree is constant during the learning process. However, the exploration-exploitation trade-off is crucial in a reinforcement learning problem. The agent has to exploit the already known actions, but it also needs to explore in order to learn new actions that may be better.

Algorithm 1 shows a variant of the classical PI^2, by integrating the covariance matrix adaptation (CAM), in which the exploration decays during learning in favour of the exploitation [18], [19]. In the early stages of the learning it is convenient to have a high exploration degree to discover the best alternatives. On the contrary, in order to exploit the learned task, the exploration should be low in the final stages of the process. For this reason, a gradual decay of the exploration level has been implemented. In particular, exploration decays during learning in accordance with the law described in line 11 of Algorithm 1, where $0 \ll \gamma < 1$, and u is the update number. The value γ depends on the number of updates required to learn the task.

The algorithm parameters used in this work are detailed in Table I. Moreover, in the considered application the parameters of the policy are the synergies coefficient $\boldsymbol{\sigma}$ and the trajectories $\boldsymbol{\tau}_k$ represent the hand configuration corresponding to the synergy coefficients by means of (2) that the hand reaches using its low-level control, i.e.

$$\boldsymbol{\tau}_k = \tau(\boldsymbol{\theta}_k) = \boldsymbol{q}(\boldsymbol{\sigma}).$$

The reward function S is based on the force closure cost

function [26] i.e.

$$S_k = S(\boldsymbol{q}(\boldsymbol{\sigma}_k)) = r(\boldsymbol{\sigma}).$$

All these parameters and functions will be better detailed in Sects. III and IV.

TABLE I
PI^2 PARAMETERS SETTING.

$N = 10$	Number of updates
$K = 5$	Number of trials per update
$\lambda = 1000$	Exploration level
$\gamma = 0.9$	Exploration decay coefficient

Algorithm 1 PI^2

Input: $\boldsymbol{\theta}$
λ^{init}
K
γ
$\Sigma_{init} = \lambda I$

while $true$ **do**
  for $k = 1$ to K **do**
    $\boldsymbol{\theta}_k \sim \mathcal{N}(\boldsymbol{\theta}, \Sigma)$
    $\boldsymbol{\tau}_k = \tau(\boldsymbol{\theta}_k)$
    $S_k \equiv S(\boldsymbol{\tau}_k)$
  end for
  for $k = 1$ to K **do**
    $P_k = \frac{e^{-\frac{1}{\lambda}S_k}}{\sum_{k=1}^{K} e^{-\frac{1}{\lambda}S_k}}$
  end for
  $\boldsymbol{\theta}^{new} = \sum_{k=1}^{K} P_k \ (\boldsymbol{\theta}_k - \boldsymbol{\theta})$
  $\Sigma = \gamma^u \Sigma_{init}$
end while

III. THE SCHUNK S5FH MOTOR SYNERGIES

The synergy-based reinforcement learning strategy has been tested experimentally on an anthropomorphic hand, the Schunk 5-Finger Hand [27], [28]. The hand motion is driven by 9 motors that move 20 joints. The majority of the joints are actuated through leadscrew mechanisms converting linear into rotational motion. The other joints are passively moved by means of a rigid linkage that realizes couplings to reproduce natural movements using a reduced number of independent degrees of freedom. Therefore, the hand has its own mechanical synergies represented by the (20×9) $\boldsymbol{S}_m$ matrix that maps the motor space into the joint space, as in the following:

$$\boldsymbol{q} = \boldsymbol{S}_m \boldsymbol{m} + \boldsymbol{q}_0, \tag{1}$$

where $\boldsymbol{q} \in \mathbb{R}^n$, with $n = 20$, is the vector of joint variables, $\boldsymbol{m} \in \mathbb{R}^m$, with $m = 9$, is the vector of motor variables and $\boldsymbol{q}_0$ is a mechanical offset characterizing joint angles when the motors position are set to zero. To further reduce the dimension of the control problem, postural synergies are mapped from human hand grasping demonstration using the results obtained in [29], [30] where the effectiveness of the first three synergies subspace in planning and control grasping actions has been demonstrated. As a result of these studies the (9×3) $\boldsymbol{S}_s$ matrix of the first three eigenvectors sorted in decreasing order of variance are computed in the motor space using Principal Component Analysis (PCA). Hence, the computed motor synergies matrix $\boldsymbol{S}_s$ and the vector $\bar{\mathbf{m}}$, that represents the origin of the synergies subspace, are connected to the hand configuration space by means of the mechanical synergies matrix $\boldsymbol{S}_m$. The mapping between the synergies subspace and the joint space is given by the following expression:

$$\boldsymbol{q} = \boldsymbol{S}_m(\boldsymbol{S}_s \boldsymbol{\sigma} + \bar{\mathbf{m}}) + \boldsymbol{q_0}, \tag{2}$$

where $\boldsymbol{\sigma}$ represents the (3×1) vector of synergy coefficients. The synergies subspace, represented by the matrix $\boldsymbol{S}_s$, has been computed using human grasp data and a mapping algorithm available from previous work [31], [32]. Because of underactuation, whatever is the mapping method of human hand motion to the robotic hand, a faithful mapping will never be achieved and part of the information will be inevitably lost. Thus, the coefficients of the first three synergies, computed by projection of a given grasp in the synergies subspace, can determine only a good hand preshaping but cannot reproduce a stable grasp [29].

IV. REWARD FUNCTION AND GRASP QUALITY

The success of a reinforcement learning algorithm based on PI^2 is the proper choice of the policy representations suitable for the particular application. Examples available in the literature of policy representation in the context of robotic manipulation are the Gaussian Mixture Model (GMM) and Gaussian Mixture Regression (GMR) used in [33] and DMPs for a compact representation of a movement [13]. In the particular application of anthropomorphic hand grasping synthesis, to improve the PI^2 performance and ensure fast convergence, the algorithm has been implemented in the synergies subspace and the learning policy is based on the synergy-based approach. Taking advantage from dimesionality reduction, the optimal policy parameters are searched directly into the synergies subspace. The parameters of the policy have a precise meaning; in particular, the vector $\boldsymbol{\theta}$ represents the postural synergy coefficients, i.e. $\boldsymbol{\theta} = \boldsymbol{\sigma}$. Therefore, each trial extracted from the multivariate Gaussian distribution is a robotic hand grasp configuration. In this framework, a synergy-based quality index $V(\boldsymbol{\sigma})$ has been used in the reward function where it is summed to a discontinuous function ϕ that drastically penalizes the reward if no contact occurred. In particular the adopted force-closure cost function has been introduced in [26]. This cost function has to be minimized to achieve the best grasp feasible with the given set of synergies.

Specifically, the reward function $r(\boldsymbol{\sigma})$ used in the PI^2 algorithm is defined by:

$$r(\boldsymbol{\sigma}) = \beta V(\boldsymbol{\sigma}) + \phi \tag{3}$$

where $\beta = 10^{-6}$ is a normalization coefficient and ϕ is:

$$\phi = \begin{cases} 0 & \text{if grasp succeeds} \\ 10^3 & \text{if grasp fails} \end{cases} \quad (4)$$

The value $\phi = 10^3$ has been chosen so high in order to penalize decisively the failed grasp, where no contact between robotic hand and object occurs.

PI^2 is a global method, yet the convergence to the global optimum is not ensured when variations to the classical version of the method are introduced, as described in Sect. II. Nevertheless, the choice of the policy and a good initialization of the parameters allows reaching an excellent solution. In this work, the initial policy parameters for each grasp is computed as the synergy coefficients corresponding to the closer object contained in the reference set used for synergies computation [30].

Starting from hand preshaping, the learning algorithm is in charge of searching autonomously for a stable grasp. The hand preshaping configuration is the result of a sort of imitation learning of human actions and contains information on task compatibility, i.e. number of fingers involved in the grasp, influencing the cost function, and object affordance. The imitation learning determines the policy and the initial policy parameters, while the policy improvement learning algorithm, based on the chosen reward function, ensures stability and adaptability to new objects and determine the task execution. In Fig. 1 a schematic representation of the strategy is reported.

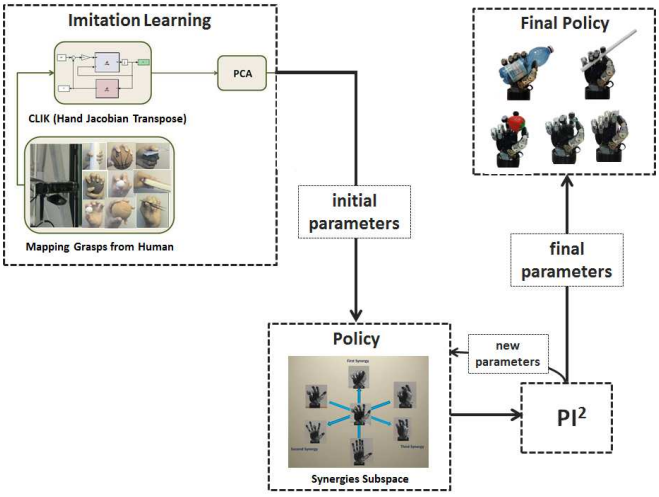

Fig. 1. A schematic representation of the learning algorithm.

V. Experiments

To experimentally show the performance of the presented method and validate the choice of the policy, the PI^2 algorithm has been tested on the SCHUNK 5-Finger Hand. The S5FH is controlled using a Robot Operating System (ROS) package that contains the driver for the low-level interface and enables an easy control of the hand using a customized library written in C++ [34]. The low-level control law is a kinematic control strategy developed in the synergies subspace where the fingers reference position is given by the output of the learning algorithm as desired synergy coefficients. Moreover, in order to limit the contact forces during the execution of the grasp, the desired target is modified on the basis of the measured motor current and of a defined threshold that is related to the texture of the object, for more details see [29]. The Robotics System Toolbox is used to provide an interface between MATLAB and ROS in such a way as to create a ROS node in MATLAB to exchange messages with the hand driver node.

TABLE II

Values of the force closure cost function for each step of the learning process depicted in Figs. 2, 3, 4, 5, 6, 7.

Object	Preshaping	Intermediate grasp	Final Grasp
Bottle	$2.48 \cdot 10^6$	$4.06 \cdot 10^5$	$1.75 \cdot 10^5$
Cylindrical object	$7.45 \cdot 10^6$	$6.23 \cdot 10^6$	$5.37 \cdot 10^6$
Card	$6.96 \cdot 10^6$	$5.46 \cdot 10^6$	$4.60 \cdot 10^6$
Strawberry	$5.52 \cdot 10^6$	$4.96 \cdot 10^6$	$2.70 \cdot 10^6$
Marble	$3.38 \cdot 10^5$	$2.57 \cdot 10^5$	$1.84 \cdot 10^5$
Needle	$2.08 \cdot 10^5$	$1.45 \cdot 10^5$	$9.37 \cdot 10^4$

The PI^2 algorithm has been implemented in MATLAB. To test the learning capacity of the algorithm in different situations, power, precision and lateral grasp have been considered as well as objects of different shape and size. The learning results are shown in Figs. 2 to 7. In each of these figures three images of the experiment with the same object are reported. The first image represents the hand in the preshaping phase that is the result of the imitation learning as shown in Fig. 1. The intermediate image represents the hand and the object during learning. To show the progress of the algorithm during the learning phase, we have chosen for each object a configuration of the hand corresponding to an intermediate trial of the algorithm. Finally, the third image represents the final hand configuration corresponding to the convergence of the algorithm and to a stable grasp. In Table II, the force closure cost value corresponding to each learning phase (arranged from left to right in order of appearance of the images in the figures) are reported. As expected, the cost value decreases from left to right.

Fig. 2. Power grasp example: bottle.

First of all it should be noted that at the end of the learning all the objects are grasped with accuracy and stability. Moreover, starting from the initial parameters of the policy, the algorithm is able to distinguish the number of fingers involved in the grasp. This result is amazing, especially considering that the hand, using only the initial parameters of the policy, cannot grasp any of the considered objects.

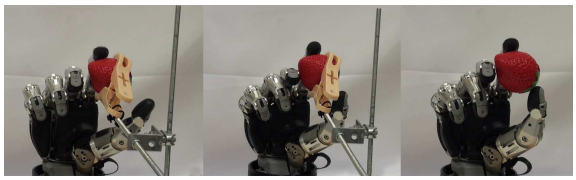

Fig. 3. Tripodal grasp example: strawberry.

Fig. 4. Bipodal grasp example: marble.

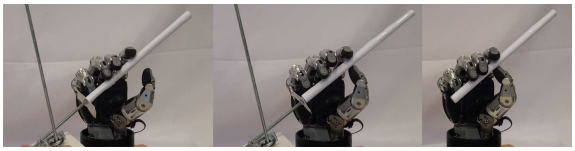

Fig. 5. Precision grasp with five fingers example: small cylinder.

Fig. 6. Lateral side grasp example: card.

The algorithm implemented in this work provides a standard number of updates for each object. In particular, the algorithm performs ten updates with five trials for each of them. However, the experimental tests have shown that the task is learned in fewer trials, as illustrated in Table III. In this work, a certain grasp is evaluated as successful when the object is not lost, and this evaluation can be easily performed on the basis of the measured values of the motor currents. It is important to emphasize that to avoid that hand configurations, generated by the learning algorithm, could exert excessively high contact forces on the object, we have introduced a current threshold in the low-level control of the hand such that beyond this threshold the motors are stopped.

VI. Discussion of the Results

A. Comparison of PI^2 performance in the full-DoF system

To show how dimensionality reduction of the policy search space makes the algorithm extremely fast and efficient, we report the results obtained using the RL algorithm in the full-DoF motor space. For the sake of comparison, two different objects have been selected to perform a power grasp and a precision grasp. When the complete motor space is considered as the search space of the policy, the convergence of the algorithm is not ensured. In Fig. 8 the results of the RL algorithm applied to a tripodal grasp are reported. The initial parameters of the search policy are chosen in the full-DoF motor space. It is possible to observe from images that the algorithm do not converge and the object is not grasped. For simpler grasps the algorithm converges but with a higher number of trials. In Fig. 9 the bottle has been grasped after 20 trials, i.e. 5 more than the trials obtained when the algorithm operates in the synergies subspace. In addition, the final grasp has a higher value of the force closure cost function, $3.01 \cdot 10^6$.

TABLE III

Minimum number of trials for a stable grasp.

Bottle	15 Trials
Card	15 Trials
Cylindrical object	20 Trials
Strawberry	25 Trials
Marble	40 Trials
Needle	45 Trials

B. Limits of the method

A limitation of this work is the absence of an arm accompanying the hand in the phase of reaching towards the object and preshaping. Thus, learning knowledge related to object affordance and task description, that are obviously two important aspects of grasping actions, will be addressed in future work where the whole hand-arm system will be considered. Nevertheless, at this stage part of the information is contained in the initial parameters of the policy. Another limitation regards the object association with the closest target in the reference data set to generate the initial parameters of the policy (i.e. the preshaping of the hand), see Sect. IV. In this regard, in future works a vision system can extract the characteristics of the object and associate them to the closest object among those contained in a database. After identifying the nearest object, the synergies coefficients for hand preshaping can be computed. Furthermore, the vision system will be utilized also to automatically evaluate the success of the grasp (Eq. (4)) for the reward function.

VII. Conclusions

In this work, a reinforcement learning algorithm has been implemented for learning grasping with an anthropomorphic robotic hand using a synergy-based search policy. In particular, the chosen search approach is the Policy Improvement with Path Integrals and comes from the field of stochastic optimization. This algorithm has been chosen for the characteristic to be very effective in the field of robotics and in particular in applications in which the dimension of the "actions" space is high. In order to ensure the convergence of the algorithm and improve the performance we have chosen, as policy, an approach based on postural synergies of the robotic hand. The convergence towards a solution that in our case corresponds to a stable grasp, confirms that the chosen reward function as the synergy-based force closure cost and the chosen exploration law is a winning choice.

Fig. 7. Precision grasp example of a very small object: needle.

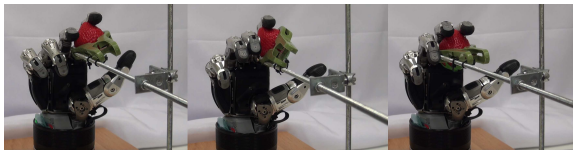

Fig. 8. Tripodal grasp example using the full-DoF search space: strawberry.

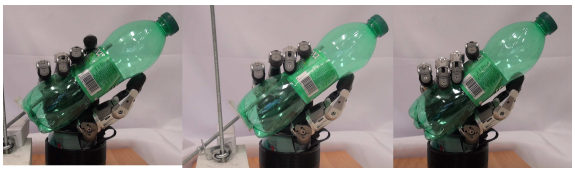

Fig. 9. Power grasp example using the full-DoF search space: bottle.

REFERENCES

[1] F. Ficuciello and B. Siciliano, "Learning in robotic manipulation: The role of dimensionality reduction in policy search methods. Comment on Hand synergies: Integration of robotics and neuroscience for understanding the control of biological and artificial hands by Marco Santello, Matteo Bianchi, Marco Gabiccini, Emiliano Ricciardi, Gionata Salvietti et al." *Physics of Life Reviews*, vol. 17, pp. 36-37, 2016.

[2] S. Schaal and C. Atkeson, "Learning control in robotics," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 20–29, 2010.

[3] F. Ficuciello, G. Palli, C. Melchiorri, and B. Siciliano, "Postural synergies and neural network for autonomous grasping: A tool for dextrous Prosthetic and Robotic Hands," in *Converging Clinical and Engineering Research on Neurorehabilitation* J.L. Pons, D. Torricelli and M. Pajaro Eds. " pp. 467–480, Springer Berlin Heidelberg, 2012.

[4] F. Ficuciello, D. Zaccara, and B. Siciliano, "Learning grasps in a synergy-based framework," *International Symposium on Experimental Robotics*, 2016.

[5] A. Sahbania, S. El-Khouryc, and P. Bidauda, "An overview of 3D object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, 2012.

[6] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. pp. 1371–1394, Springer, 2008.

[7] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survay of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[8] Y. Li, J. Fu, and N. Pollard, "Data-driven grasp synthesis using shape matching and task-based pruning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 4, pp. 732–747, 2007.

[9] R. Pelossof, A. Miller, P. Allen, and T. Jebara, "An SVM learning approach to robotic grasping," in *Proc. IEEE International Conference on Robotics and Automation*, New Orleans, USA, 2004, pp. 3512–3518.

[10] P. Kormushev, S. Calinon, and D. Caldwell, "Reinforcement learning in robotics: Applications and real-world challenges," *Robotics*, vol. 2, pp. 122–148, 2007.

[11] J. Peters and S. Schaal, "Policy gradient methods for robotics," in *Proc. IEEE/RSJ International Conference on Intelligent Robotics Systems*, Beijing, China, 2006, pp. 2219–2225.

[12] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 1992.

[13] J. Kober and J. Peters, "Learning motor primitives for robotics," in *Proc. IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 2112–2118.

[14] E. Theodorou, J. Buchli, and S. Schaal, "Learning policy improvements with path integrals," *Journal of Machine Learning Research*, vol. 9, pp. 828–835, 2010.

[15] F. Stulp, A. Theodorou, and S. Schaal, "Reinforcement learning with sequences of motion primitives for robust manipulation," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1360–1370, 2012.

[16] R. Rubinstein and D. Kroese, *The Cross-Entropy Method: A unified approach to combinatorial optimization, Monte-Carlo simulation and Machine Learning*. Springer-Verlag: New York, NY, 2004.

[17] N. Hansen, "The CMA evolution strategy: A comparing review," *Towards a New Evolutionary Computation*, J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, Eds. pp. 75–102, Springer: Berlin, Germany, 2006.

[18] F. Stulp and O. Sigaud, "Path integral policy improvement with covariance matrix adaptation," in *Proc. 29th International Conference on Machine Learning*, Edinburgh, Scotland, 2012, pp. 2112–2118.

[19] E. Theodorou, J. Buchli, and F. Schaal, "A generalized path integral control approach to reinforcement learning," *Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, 2010.

[20] P. Kormushev, S. Calinon, and D. Caldwell, "Robot motor skill coordination with em-based reinforcement learning," in *Proc. IEEE/RSJ International Conference on Intelligent Robotics Systems*, Taipei, Taiwan,, 2010, pp. 3232–3237.

[21] P. K. B. Ugurlu, S. Calinon, N. Tsagarakis, and D. Caldwell, "Bipedal walking energy minimization by reinforcement learning with evolving policy parameterization." in *Proc. IEEE/RSJ International Conference on Intelligent Robotics Systems*, San Francisco, CA, USA, 2011, pp. 318–324.

[22] F. Stulp, "Adaptive exploration for continual reinforcement learning," in *Proc. IEEE/RSJ International Conference on Intelligent Robotics Systems*, Vilamoura, Algarve, Portugal, 2012, pp. 318–324.

[23] N. G. Tsagarakis, G. Metta, G. Sandini, D. Vernon, R. Beira, F. Becchi, L. Righetti, J. Santos-Victor, A. J. Ijspeert, M. Carrozza, and D. Caldwell, "iCub: The design and realization of an open humanoid platform for cognitive and neuroscience research," *Advanced Robotics*, vol. 21, pp. 1151–1175, 2007.

[24] E. Rombokas, M. Malhotra, E. Theodorou, E. Todorov, and Y. Matsuoka, "Reinforcement learning and synergistic control of the act hand," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 2, pp. 569–577, 2012.

[25] M. Santello, M. Flanders, and J. Soechting, "Postural hand synergies for tool use," *Journal of Neuroscience*, vol. 18, no. 23, pp. 10 105–10 115, 1998.

[26] A. Bicchi, "On the closure properties of robotic grasping," *International Journal of Robotics Research*, vol. 14, no. 4, pp. 319–334, 1994.

[27] "SCHUNK Hand webpage," http://mobile.schunk-microsite.com/en/produkte/produkte/servoelektrische-5-finger-greifhand-svh.html.

[28] S. Ruehl, C. Parliz, G. Heppner, A. Hermann, A. Roennau, and R. Dillman, "Experimental evaluation of the SCHUNK 5-Finger gripping hand for grasping tasks," in *Proc. IEEE Int. Conf. on Robotics and Biomimetics*, Bali, Indonesia, 2014, pp. 2465–2470.

[29] F. Ficuciello, A. Federico, V. Lippiello, and B. Siciliano, "Synergies evaluation of the SCHUNK S5FH for grasping control," in *15th International Symposium on Advances in Robot Kinematics*, 2016.

[30] F. Ficuciello, G. Palli, C. Melchiorri, and B. Siciliano, "Postural synergies of the ub hand iv for human-like grasping," *Robotics and Autonomous Systems*, vol. 62, pp. 357–362, 2014.

[31] G. Palli, C. Melchiorri, G. Vassura, U. Scarcia, L. Moriello, G. Berselli, A. Cavallo, G. D. Maria, C. Natale, S. Pirozzi, C. May, F. Ficuciello, and B. Siciliano, "The DEXMART hand: Mechatronic design and experimental evaluation of synergy-based control for human-like grasping," *International Journal of Robotics Research*, vol. 33, pp. 799–824, 2014.

[32] F. Ficuciello, G. Palli, C. Melchiorri, and B. Siciliano, "Experimental evaluation of postural synergies during reach to grasp with the UB Hand IV," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, 2011, pp. 1775–1780.

[33] F. Guenter, M. Hersch, S. Calinon, and A. Billard, "Reinforcement learning for imitating constrained reaching movements," *Advanced Robotics*, vol. 21, pp. 1521–1544, 2007.

[34] "schunk_svh_driver," http://wiki.ros.org/schunk_svh_driver.